## AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

## Listing of Claims:

1. (currently amended) A method for processing data using a programmable processor comprising:

decoding a single instruction for writing data to memory based on a mask and data contained in at least one register ~~specifying both a mask and a register containing data~~, the mask comprising fields that each correspond to a field of the data contained in the at least one register;

detecting some of the fields of the mask as having a predetermined value ~~and identifying~~ to identify corresponding fields of the data contained in the at least one register as write-enabled data fields; and

writing the write-enabled data fields to a specified memory location.

2. (original) The method of claim 1 wherein each of the fields of the mask has a width of one bit.

3. (currently amended) The method of claim 1 wherein each of the fields of the data contained in the at least one register has a width of one bit.

4. (original) The method of claim 1 wherein the writing step further comprises reading an unaltered field of data from the specified memory location and writing the unaltered field of data along with the write-enabled data fields to the specified memory location.

5. (original) The method of claim 1 wherein the mask is contained in a specified register.

6. (currently amended) The method of claim 1 wherein the memory location is ~~contained in a specified~~ specified by a register.

7. (original) The method of claim 1 wherein the specified memory location comprises a section of memory having a specific width and beginning at a specific memory address.

8. (original) The method of claim 1 wherein the predetermined value is a logic 1.

9. (currently amended) The method of claim 1 further comprising:

decoding a second single instruction specifying a ~~third and a fourth~~ register ~~each~~ containing a first plurality of floating-point operands and another register containing a second plurality of floating-point operands;

multiplying the first plurality of ~~floating point~~ floating-point operands ~~in the third register~~ by the second plurality of floating-point operands ~~in the fourth register~~ to produce a plurality ~~or~~ of products;

and providing the plurality of products to partitioned fields of a result register as a catenated result.

10. (currently amended) A computer-readable storage medium having stored therein a plurality of instructions that cause a ~~computer~~ programmable processor to perform data operations:

at least some of the instructions including a ~~store multiplex~~ single instruction for selectively storing data ~~in a programmable processor~~, the ~~store multiplex~~ single instruction capable of instructing ~~a computer~~ the programmable processor to perform operations comprising:

decoding the ~~store multiplex~~ single instruction to obtain a mask and data contained in at least one register ~~specifying both a mask and a register containing data~~, the mask comprising fields that each correspond to a field of the data contained in the at least one register;

detecting some of the fields of the mask as having a predetermined value ~~and identifying~~ to identify corresponding fields of the data contained in the at least one register as write-enabled data fields; and

writing the write-enabled data fields to a specified memory location.

11. (previously presented) The computer-readable storage medium of claim 10 wherein each of the fields of the mask has a width of one bit.

12. (currently amended) The computer-readable storage medium of claim 10 wherein each of the fields of the data contained in the at least one register has a width of one bit.

13. (previously presented) The computer-readable storage medium of claim 10 wherein the writing step further comprises reading an unaltered field of data from the specified memory location and writing the unaltered field of data along with the write-enabled data fields to the specified memory location.

14. (previously presented) The computer-readable storage medium of claim 10 wherein the mask is contained in a specified register.

15. (currently amended) The computer-readable storage medium of claim 10 wherein the memory location is ~~contained in a specified~~ specified by a register.

16. (previously presented) The computer-readable storage medium of claim 10 wherein the specified memory location comprises a section of memory having a specific width and beginning at a specific memory address.

17. (previously presented) The computer-readable storage medium of claim 10 wherein the predetermined value is a logic 1.

18. (currently amended) The computer-readable storage medium of claim 10 wherein at least some of the instructions further include a group ~~floating-point~~ floating-point multiply instruction for multiplying ~~floating-point~~ floating-point data in ~~a~~ the programmable processor, the group ~~floating-point~~ floating-point multiply instruction capable of instructing the ~~computer~~ programmable processor to perform operations comprising:

decoding the group ~~floating-point~~ floating-point multiply instruction specifying a ~~third and a fourth~~ register ~~each~~ containing a first plurality of floating-point operands and another register containing a second plurality of floating-point operands;

multiplying the first plurality of ~~floating-point~~ floating-point operands ~~in the third register~~ by the second plurality of floating-point operands ~~in the fourth register~~ to produce a plurality ~~or~~ of products; and

providing the plurality of products to partitioned fields of a result register as a catenated result.

19. - 27. (canceled)

28. (currently amended) A method for processing data in a programmable processor, the method comprising:

decoding a single instruction for performing a bitwise insert operation on data in ~~registers~~ at least one register in a register file within the programmable processor, the bitwise insert operation operating on a first operand and a second operand stored in ~~registers~~ the at least one register in the register file; and

for each bit in the first operand, the bitwise insert operation inserting the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has a first predetermined value.

29. (previously presented) The method of claim 28 wherein the first predetermined value is a logic 1.

30. (previously presented) The method of claim 28 wherein for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has a second predetermined value.

31. (previously presented) The method of claim 30 wherein the second predetermined value is a logic 0.

32. (previously presented) The method of claim 28 further comprising a step of storing the destination value into memory.

33. (previously presented) The method of claim 28 wherein each of the first and second operands has a width of 64 bits.

34. (currently amended) The method of claim 28 further comprising a step of executing a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of ~~an operand register~~ registers in the register file ~~plurality of registers~~ to produce a catenated result that is returned to a register in the register file ~~plurality of registers~~, wherein the catenated result comprises a plurality of individual floating-point results.

35. (currently amended) A computer-readable storage medium having stored therein a plurality of instructions that cause a ~~computer~~ programmable processor to perform operations on

data ~~stored in registers~~ in the ~~computer~~ programmable processor, the plurality of instructions comprising:

an instruction that causes the processor to perform a bitwise insert operation on data in at least one ~~registers~~ in a register file within the programmable processor, the bitwise insert operation operating on a first operand and a second operand stored in ~~registers~~ the at least one register in the register file; and

wherein for each bit in the first operand, the bitwise insert operation inserts the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has a first predetermined value.

36. (previously presented) The computer-readable storage medium of claim 35 wherein the first predetermined value is a logic 1.

37. (previously presented) The computer-readable storage medium of claim 35 wherein for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has a second predetermined value.

38. (previously presented) The computer-readable storage medium of claim 37 wherein the second predetermined value is a logic 0.

39. (previously presented) The computer-readable storage medium of claim 35 wherein the destination value is stored into memory.

40. (previously presented) The computer-readable storage medium of claim 35 wherein each of the first and second operands has a width of 64 bits.

41. (currently amended) The computer-readable storage medium of claim 35 wherein the plurality of instructions further comprises a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of ~~an operand register~~ registers in the register file ~~plurality of registers~~ to produce a catenated result that is returned to a register in the register file ~~plurality of registers~~, wherein the catenated result comprises a plurality of individual floating-point results.